

Northumbria Research Link

Citation: Khelifi, Fouad (2018) On the Security of a Stream Cipher in Reversible Data Hiding Schemes Operating in the Encrypted Domain. Signal Processing, 143. pp. 336-345. ISSN 0165-1684

Published by: Elsevier

URL: <https://doi.org/10.1016/j.sigpro.2017.09.020>
<<https://doi.org/10.1016/j.sigpro.2017.09.020>>

This version was downloaded from Northumbria Research Link:
<http://nrl.northumbria.ac.uk/id/eprint/32057/>

Northumbria University has developed Northumbria Research Link (NRL) to enable users to access the University's research output. Copyright © and moral rights for items on NRL are retained by the individual author(s) and/or other copyright owners. Single copies of full items can be reproduced, displayed or performed, and given to third parties in any format or medium for personal research or study, educational, or not-for-profit purposes without prior permission or charge, provided the authors, title and full bibliographic details are given, as well as a hyperlink and/or URL to the original metadata page. The content must not be changed in any way. Full items must not be sold commercially in any format or medium without formal permission of the copyright holder. The full policy is available online: <http://nrl.northumbria.ac.uk/policies.html>

This document may differ from the final, published version of the research and has been made available online in accordance with publisher policies. To read and/or cite from the published version of the research, please visit the publisher's website (a subscription may be required.)



**Northumbria
University**
NEWCASTLE



UniversityLibrary

On the Security of a Stream Cipher in Reversible Data Hiding Schemes Operating in the Encrypted Domain

Fouad Khelifi¹

Abstract

Reversible data hiding in encrypted images has recently emerged as an effective approach to embed and extract a message in the encrypted domain and losslessly recover the host data while maintaining its confidentiality through encryption. That is, the data hider can embed and extract additional data without knowing the image. This approach can be used in cloud applications where the service provider, i.e., the data hider, is not authorized to access the visual content of the host data for security and privacy purposes. Most existing techniques that have been reported in the literature apply a bit-wise encryption method, also known as the stream cipher, prior to data hiding. However, because of the spatial redundancy that characterizes natural images, the security of such an encryption could be compromised. This work is the first one that analyzes reversible data hiding in encrypted images from a security perspective. It proposes a Ciphertext-Only Attack (COA) and highlights the weakness of current state-of-the-art data hiding systems in the encrypted domain. We particularly show how the data hider can break the security of the encryption system and consequently discloses the visual content of encrypted images. Finally, possible solutions to combat COA with existing systems are discussed.

Keywords: Reversible data hiding, Encryption, Security, Inter-pixel redundancy.

1. Introduction

Reversible Data Hiding (RDH) was first introduced in the pixel domain with the aim of embedding as much data as possible in the host image provided that the distortion incurred is visually acceptable [1, 2, 3]. Reversibility suggests that the decoder can recover the original host image losslessly. The idea of RDH in encrypted images is inspired by existing works on information processing in the encrypted domain for cloud computing and privacy-preserving applications such as data compression [4, 5], watermarking [6] and signal representation and transforms [7, 8]. The idea has also been extended in [9] for RDH in encrypted compressed bitstream. As the name suggests, RDH in the encrypted domain consists of embedding and extracting the message in the encrypted domain [10, 11]. This ensures the confidentiality of the host image through encryption since the data hider can only embed and extract additional data without access to the visual content of the image. Such systems can be used in applications where the content is protected from the service provider (i.e. the data hider) while being shared with authorized users.

RDH techniques in the encrypted domain can be cast in two classes based on whether the embedding room is created before or after encryption [12]. The very first attempts in the literature create space after encryption [10, 11, 13]. This approach has also been broadly adopted in recent years [14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. In the other category

¹Department of Computer and Information Sciences, Northumbria University, NE2 1XE, UK. E-mail: fouad.khelifi@northumbria.ac.uk

of RDH in encrypted images, the embedding space is created before encryption [12, 24, 25, 26]. Because this approach attempts to create room in the spatial domain, inter-pixel redundancy can be exploited in an efficient way for higher embedding capacity. It is also worth mentioning an interesting feature in some systems, called separability, which has been introduced in [14] for potential use in areas where the extraction of additional data needs to be performed *separately* from image recovery [14, 12, 24, 26, 15, 17, 22].

With the exception of a single attempt, reported recently in [21], all existing data hiding techniques in the encrypted domain have been assessed on the basis of two conflicting measures, i.e., the capacity of data hiding and the quality of marked images after decryption. Thus, the security aspect has been absent in existing works on RDH in encrypted images. Nonetheless, the security of such systems is paramount because the primary aim of encryption is to protect the image content from the data hider. The aforementioned paper that has touched on the security of a RDH system in encrypted images considered the Watermarked Only Attack (WOA) in which the attacker has access only to the encrypted marked image [21]. However, the problem where the attacker has access to the encrypted image before data hiding was never addressed. Note that this attack, which is often referred to as the ciphertext-only attack, is very likely to occur since the data hider can act as the malicious attacker. It is worth noting that the ciphertext-only attack has been used in [27] to demonstrate the weakness of a selective image scrambling scheme. It has also been successful in breaking a number of permutation-only image encryption systems [28, 29] though its performance remains clearly lower than that of the known-plaintext attack and the chosen-plaintext attack [30, 31, 32]. In this context, it is also worth mentioning other attempts that break the security of image-based systems when the key is re-used multiple times [33, 34, 27, 32, 35].

This paper analyzes the security of a stream cipher that has been widely applied in current state-of-the-art RDH techniques operating in the encrypted domain. From the aforementioned techniques, the stream cipher was particularly used in [11, 14, 13, 15, 12, 16, 17, 18, 21, 22, 23, 25, 26]. Because the primary aim of encryption in such RDH schemes is to protect the content of images from the data hider, the cipher security is of paramount significance for privacy-preserving applications. Compared with other image encryption techniques, the stream cipher is faster and provides more flexibility for the data hider to create room and embed additional data. However, because of the spatial redundancy that characterizes natural images, the security of such a cipher could be compromised. This paper proposes a ciphertext-only attack and highlights the weakness of current state-of-the-art data hiding systems in the encrypted domain. A new algorithm is proposed to estimate the stream cipher key and break the security of the encryption system. As a result, the estimated key enables the data hider to disclose the visual content of any encrypted image. The novelty of this work is twofold: (i) A cryptanalysis on the stream cipher in RDH systems is provided. Unlike existing work, this paper evaluates RDH systems in the encrypted domain from a *security* point of view where the data hider acts as the attacker to disclose the image content. (ii) A new ciphertext-only attack exploiting inter-pixel redundancy is proposed. The technique can efficiently estimate the stream cipher key and consequently disclose the visual content of stream cipher encrypted images.

The rest of the paper is structured as follows. Section 2 presents a formulation of the security problem with current schemes of reversible data hiding in encrypted images. In section 3, the proposed technique for breaking the stream cipher

key is described. Section 4 demonstrates the performance of the proposed technique through an experimental analysis on natural images. Section 5 discusses possible solutions to tackle the security problem with existing RDH in encrypted images. Finally, section 6 summarizes the contributions and concludes the paper.

2. Problem formulation

Practical scenarios of RDH in encrypted images have been suggested in [11, 12, 18, 22, 25] for secret image sharing and secure cloud computing. As illustrated by Fig.1(a), sensitive visual data such as biometrics (face, fingerprint, iris, etc.) and medical images need to be encrypted before they are sent to the cloud for storage. This ensures the confidentiality of the data since only authorized cloud users can disclose the content of images via decryption and image recovery. On the other hand, the Cloud Service Provider (CSP) may embed some meta data about the owner and the date of upload in encrypted images to facilitate their management in the database (i.e. storage, search, authentication, and retrieval). As

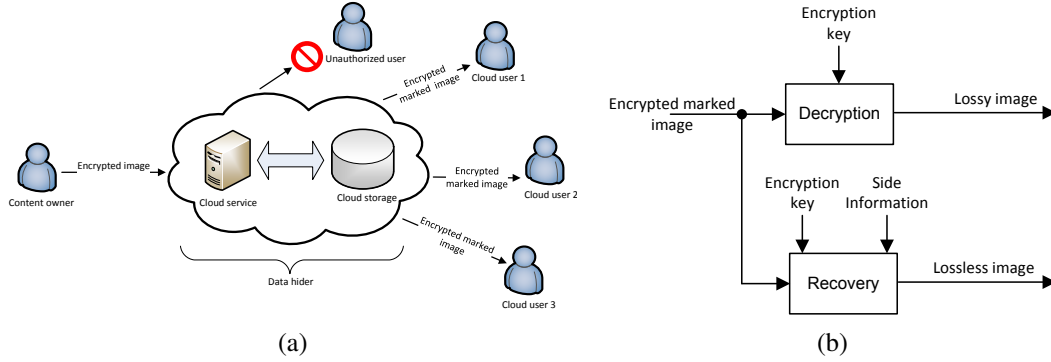


Figure 1: Application of RDH in encrypted images. (a) Secret image sharing through the cloud using RDH in encrypted images. (b) Reconstruction of the original image by the cloud user.

a result, the CSP acts as a data hider by embedding meta data in encrypted images and, upon request, serves authorized users with encrypted marked images. Therefore, the design suggests that the authorized users must be able to reconstruct the original image from the encrypted marked version. In the literature, the reported RDH schemes that operate in the encrypted domain normally offer both lossy and lossless image reconstruction depending on the key and information available at the receiver side [11, 13, 14, 15, 16, 17, 18, 20, 12, 24, 25, 26, 21]. In fact, if the encrypted marked image is directly decrypted, the reconstructed image will be similar to the original but with some loss of information due to the data embedding process. Existing research works consider two conflicting design requirements in this case. (i) The reconstruction quality, often measured by PSNR, on one hand, and (ii) The data hiding capacity, measured by the number of bits embedded, on the other hand. In addition to the encryption key, if the receiver also has access to some side information about the data hiding process (i.e., method used and location of bits altered) he can recover the image *losslessly*. This is shown in Fig.1(b). In short, secure and efficient data storage in the cloud involves two separate systems operating in cascade, i.e., the encryption system, used by the content owner, on one hand and the data hiding system, used by the CSP, on the other hand. The embedding room could be created before or after encryption (see Fig. 2). However, as pointed out earlier, none of the aforementioned papers considered the robustness of RDH in encrypted images

thoroughly from a *security* perspective. Obviously, the encryption process is of crucial importance here because the data

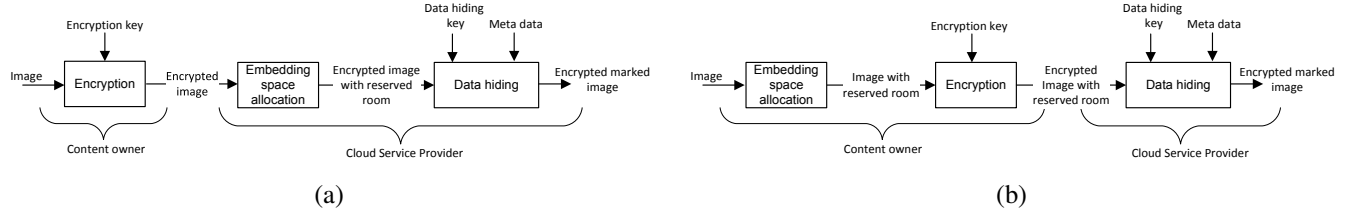


Figure 2: Embedding room allocation in RDH in encrypted images. (a) Reserving room after encryption. (b) Reserving room before encryption.

hider is not meant to access the image content. This paper is the first one to analyze RDH schemes in the encrypted domain from a *security* point of view where image content protection from the CSP is considered.

In most state-of-the-art techniques [11, 14, 13, 15, 12, 16, 17, 18, 21, 25, 26], the stream cipher is adopted to protect the image content. Denote by X^k an image to be encrypted and transmitted by the content owner where $k = 1, \dots, \ell$ and ℓ is the number of received images. Each pixel is supposedly encoded with 8 bits. The representation bit for a given pixel $X^k(i, j)$ at (i, j) can be given as

$$b_q^k(i, j) = \left\lfloor \frac{X^k(i, j)}{2^q} \right\rfloor \bmod 2, \quad q = 0, 1, \dots, 7. \quad (1)$$

Note that the content owner uses the same key to encrypt all images. The encrypted version of X^k , denoted here by \bar{X}^k , can be obtained as

$$\bar{X}^k(i, j) = \sum_{q=0}^7 \bar{b}_q^k(i, j) 2^q, \quad (2)$$

where

$$\bar{b}_q^k(i, j) = b_q^k(i, j) \oplus \Phi_q(i, j). \quad (3)$$

$\Phi_q(i, j)$ represents the key stream used for the corresponding bit plane significance level q . As will be pointed out later, the key stream is assumed to be generated from a secret key independently of the ciphertext in this paper. Given $\Phi_q(i, j)$, the adversary can inverse the process in (3) and obtain the original image as

$$b_q^k(i, j) = \bar{b}_q^k(i, j) \oplus \Phi_q(i, j). \quad (4)$$

In this paper, the data hider acts as the adversary and attempts to estimate the key stream $\Phi_q(i, j)$ as illustrated by Fig. 3. Observe that the adversary has access to ℓ encrypted images. This type of attacks is often referred to as the *Ciphertext-Only Attack* (COA) in the literature [36]. It is, however, worth mentioning that, in the case of digital images, the adversary only requires partial access to the key since the least significant bit planes (i.e., $q \in \{0, 1, 2\}$) are of low influence on the reconstruction quality as will be shown in section 4. Therefore, the adversary's primary aim is to ensure that the most significant portion of the keystream, i.e., Φ_q with $q \geq 3$ can be estimated in order to disclose the visual content of encrypted images with high quality. The proposed COA estimates the most significant portion of the key. This is described

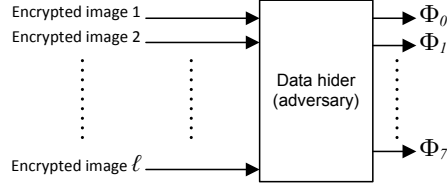


Figure 3: Ciphertext-only attack.

in the next section.

Remark: In (3), it is claimed that the key stream is generated independently of the ciphertext and this is referred to as the *synchronous* stream cipher in the literature [37]. In *asynchronous* stream ciphers, however, the ciphertext could also be involved in the key stream generation for secret sharing applications. More precisely, each bit in the key stream is generated by using the key and a number t of previous ciphertext bits. Nevertheless, the asynchronous stream cipher was not adopted in existing RDH schemes that operate in the encrypted domain since the same key stream was applied on the encrypted *marked* image at the receiver side (see for instance [11, 14, 13, 12, 16, 22, 15, 18, 23, 26, 17, 25, 21]). This can be possible only if the key stream is independent of the ciphertext because the received ciphertext is a modified version of the original one due to the data hiding process. Furthermore, the fact that the theoretical estimation of the error, caused by data hiding, was considered equal to the one obtained after direct decryption in existing works (e.g. [11, 14, 18]) suggests that there was no error propagation at the decryption stage. This is true only when the synchronous stream cipher is used. Finally, the reason for not using the asynchronous stream cipher in current RDH systems operating in encrypted images is related to design requirements as follows.

- *Directly decrypted image quality:* Since the encrypted image gets modified by the data hider, the quality of the directly decrypted image (i.e., the lossy image as shown in Fig. 1(b)) could be significantly damaged due to error propagation. In fact, because of the data hiding process, for each changed bit in the encrypted image, up to t incorrect bits would appear in the directly decrypted image when compared to the original one if the asynchronous cipher was used.
- *Reversibility:* Most state-of-the-art systems (e.g. [11, 13, 12, 22, 16, 23, 21, 25]) make use of the directly decrypted image, which contains additional data, to recover the original one. The design suggests that the changes caused by the data hiding process in the encrypted domain should equally appear at the same locations, after direct decryption, in the spatial domain. If the asynchronous stream cipher was used, such changes would spread in the spatial domain due to error propagation and, consequently, the systems would not be reversible.

3. Proposed COA

The proposed COA relies on the inter-pixel redundancy that exists in natural images. Let $M \times N$ be the size of each of the images received in encrypted form. Note that this also represents the size of the key Φ_q . For an image X^k , denote

by $P_h^k(q)$ and $P_v^k(q)$ the probability that two adjacent pixels in the horizontal and vertical direction, respectively, have the same binary value in the q^{th} bit plane significance level. That is

$$\begin{aligned} P_h^k(q) &= \text{Prob}(b_q^k(i, j) = b_q^k(i, j+1)) \\ &\quad i \in \{1, \dots, M\}, j \in \{1, \dots, N-1\} \\ &= \frac{1}{M(N-1)} \sum_{i=1}^M \sum_{j=1}^{N-1} \Gamma_{h,q}^k(i, j). \end{aligned} \quad (5)$$

where

$$\Gamma_{h,q}^k(i, j) = \begin{cases} 1 & \text{if } b_q^k(i, j) = b_q^k(i, j+1), \\ 0 & \text{Otherwise.} \end{cases} \quad (6)$$

Likewise, we have

$$\begin{aligned} P_v^k(q) &= \text{Prob}(b_q^k(i, j) = b_q^k(i+1, j)) \\ &\quad i \in \{1, \dots, M-1\}, j \in \{1, \dots, N\} \\ &= \frac{1}{(M-1)N} \sum_{i=1}^{M-1} \sum_{j=1}^N \Gamma_{v,q}^k(i, j). \end{aligned} \quad (7)$$

where

$$\Gamma_{v,q}^k(i, j) = \begin{cases} 1 & \text{if } b_q^k(i, j) = b_q^k(i+1, j), \\ 0 & \text{Otherwise.} \end{cases} \quad (8)$$

To illustrate the inter-pixel redundancy in natural images, Fig. 4 and Fig. 5 show the previously defined horizontal and vertical probabilities, respectively, on a number of standard 512×512 images. As can be seen, the redundancy increases against the bit plane significance level q . Observe that for any significant bit plane, especially for $q \geq 3$, the representation bits of two adjacent pixels are likely to be equal. Based on this observation, our technique consists of three stages: *Horizontal estimation*, *Vertical estimation*, and *Refinement*. This is described in the following. For a bit plane level q , let us assume that the key bit at the top left corner $\Phi_q(1, 1)$ is known. The estimation of $\Phi_q(1, 1)$ will be covered later in the refinement stage. As illustrated by Fig. 6, the process of estimating the remaining key bits is composed of the horizontal estimation stage and the vertical estimation stage.

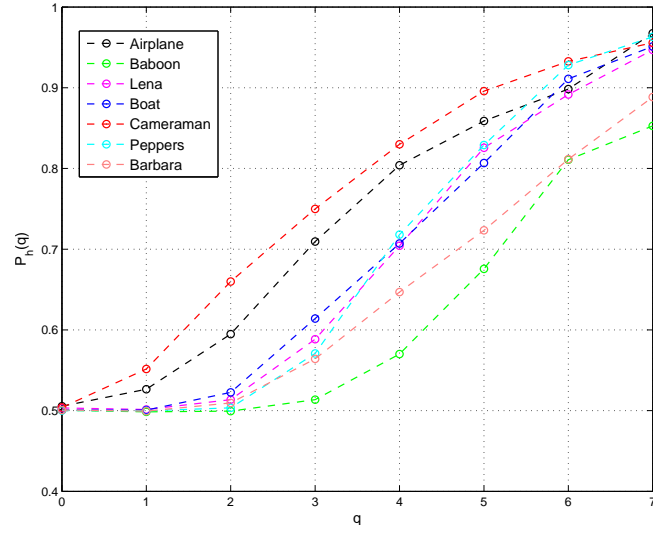


Figure 4: Inter-pixel redundancy in the horizontal direction.

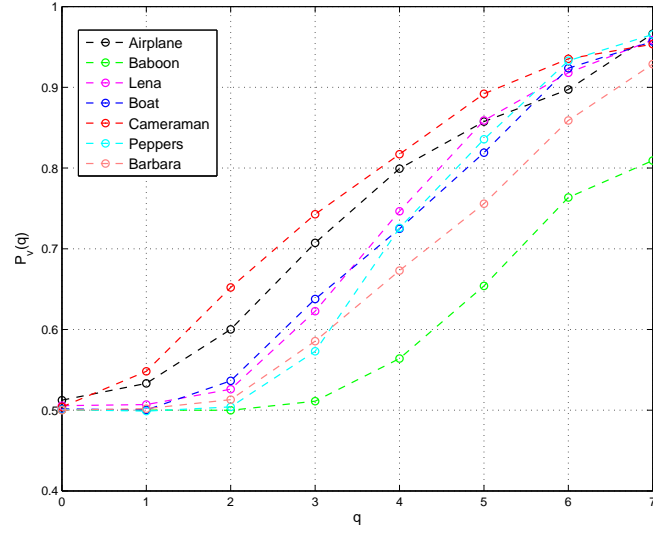


Figure 5: Inter-pixel redundancy in the vertical direction.

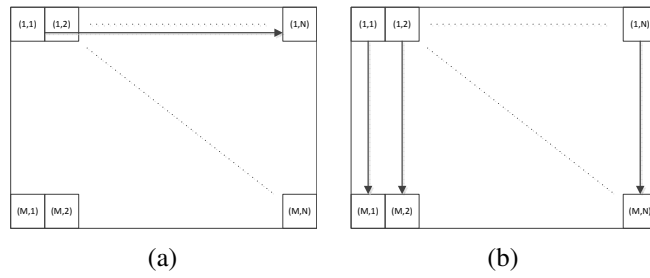


Figure 6: Estimation of the key at a given bit plane level via two stages. (a) Estimation of the key bits of the first row in the horizontal direction. (b) Estimation of the remaining bits in the vertical direction.

3.1. Horizontal estimation

In the first stage, the horizontal key bits in the first row are deduced as

$$\hat{\Phi}_q(1, j+1) = \begin{cases} \hat{\Phi}_q(1, j) & \text{if } P_h^{1,j}(q) > 0.5, \\ \langle \hat{\Phi}_q(1, j) \rangle & \text{Otherwise.} \end{cases} \quad (9)$$

where $\langle \cdot \rangle$ represents the bit flipping operation and

$$\begin{aligned}
P_h^{1,j}(q) &= \text{Prob}(\bar{b}_q^k(1,j) = \bar{b}_q^k(1,j+1)) \\
&\quad k \in \{1, 2, \dots, \ell\} \\
&= \frac{1}{\ell} \sum_{k=1}^{\ell} \Gamma_{h,q}^k(1,j).
\end{aligned} \tag{10}$$

This is the probability that two adjacent pixels at $(1, j)$ and $(1, j+1)$, respectively, have the same binary value in a bit plane at level q .

3.2. Vertical estimation

Once the key bits in the first row are estimated, the remaining bits can be deduced in the vertical direction as

$$\hat{\Phi}_q(i+1, j) = \begin{cases} \hat{\Phi}_q(i, j) & \text{if } P_v^{i,j}(q) > 0.5, \\ \langle \hat{\Phi}_q(i, j) \rangle & \text{Otherwise,} \end{cases} \tag{11}$$

where

$$\begin{aligned}
P_v^{i,j}(q) &= \text{Prob}(\bar{b}_q^k(i, j) = \bar{b}_q^k(i+1, j)) \\
&\quad k \in \{1, 2, \dots, \ell\} \\
&= \frac{1}{\ell} \sum_{k=1}^{\ell} \Gamma_{v,q}^k(i, j).
\end{aligned} \tag{12}$$

It is worth noting that the estimation of the key relies on a large number of encrypted images, denoted here by ℓ , for high accuracy. This will be addressed in section 4.

3.3. Refinement

As mentioned earlier, $\Phi_q(1, 1)$ has been assumed to be known in order to deduce the remaining bits. Obviously, two possible values for $\hat{\Phi}_q(1, 1)$ can be used in each bit plane level, i.e., $\hat{\Phi}_q(1, 1) \in \{0, 1\}$. Under the assumption that the previous horizontal and vertical estimation is accurate, the complete bit plane $\hat{\Phi}_q$ would be the flipped version of the correct one Φ_q if $\hat{\Phi}_q(1, 1)$ was incorrect. That is,

$$\hat{\Phi}_q(1, 1) = \langle \Phi_q(1, 1) \rangle \Rightarrow \hat{\Phi}_q(i, j) = \langle \Phi_q(i, j) \rangle. \tag{13}$$

Since $q = \{0, 1, \dots, 7\}$, there are 256 possibilities. One solution to find the correct key would be empirical by decrypting an image with all the possible keys in order to analyze the decrypted versions and pick up the key that leads to the most acceptable image quality. However, an alternative to deduce the right key theoretically is presented here. Note from (4)

that the flipping of a bit plane, at level q , of the key Φ_q causes a flipped bit plane of the decrypted image as

$$\langle b_q^k(i, j) \rangle = \bar{b}_q^k(i, j) \oplus \langle \Phi_q(i, j) \rangle. \quad (14)$$

Let α_s be the key that is derived from the initial 8 bit sequence $s = \{\hat{\Phi}_0(1, 1), \hat{\Phi}_1(1, 1), \dots, \hat{\Phi}_7(1, 1)\}$. Denote by $X_{s,h}^k$ a shifted version of the decrypted image X_s^k in the horizontal direction with the key α_s . Likewise, $X_{s,v}^k$ represents its shifted version in the vertical direction, i.e.,

$$X_{s,h}^k(i, j) = X_s^k(i, j + 1 \bmod N). \quad (15)$$

And,

$$X_{s,v}^k(i, j) = X_s^k(i + 1 \bmod M, j). \quad (16)$$

Let $C(X, Y)$ be the correlation coefficient given for two variables X and Y by [38]

$$C(X, Y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}, \quad (17)$$

where μ_X represents the statistical mean of X and σ_X is the corresponding standard deviation. Denote by $\langle X_s^k \rangle$ a version derived from an image X_s^k by flipping all the bit planes in its binary representation. According to (14), we have

$$\langle X_s^k \rangle = X_{\langle s \rangle}^k. \quad (18)$$

Let us define a cross-correlation measure as

$$R(X_s^k, X_{s,h}^k, X_{s,v}^k) = C(X_s^k, X_{s,h}^k) + C(X_s^k, X_{s,v}^k). \quad (19)$$

Proposition 1. $R(X_s^k, X_{s,h}^k, X_{s,v}^k) = R(\langle X_s^k \rangle, \langle X_{s,h}^k \rangle, \langle X_{s,v}^k \rangle)$.

Proof. The flipped version of X^k can be given as

$$\langle X_s^k(i, j) \rangle = 255 - X_s^k(i, j). \quad (20)$$

Hence, the statistical mean and standard deviation of $\langle X_s^k \rangle$ satisfy $\mu_{\langle X_s^k \rangle} = 255 - \mu_{X_s^k}$ and $\sigma_{\langle X_s^k \rangle} = \sigma_{X_s^k}$, respectively.

Similarly, we have $\mu_{\langle X_{s,h}^k \rangle} = 255 - \mu_{X_{s,h}^k}$, $\sigma_{\langle X_{s,h}^k \rangle} = \sigma_{X_{s,h}^k}$ and $\mu_{\langle X_{s,v}^k \rangle} = 255 - \mu_{X_{s,v}^k}$, $\sigma_{\langle X_{s,v}^k \rangle} = \sigma_{X_{s,v}^k}$. The correlation

coefficient $C(\langle X_s^k \rangle, \langle X_{s,h}^k \rangle)$ can therefore be calculated as

$$\begin{aligned}
C(\langle X_s^k \rangle, \langle X_{s,h}^k \rangle) &= \frac{E[(\langle X_s^k \rangle - \mu_{\langle X_s^k \rangle})(\langle X_{s,h}^k \rangle - \mu_{\langle X_{s,h}^k \rangle})]}{\sigma_{\langle X_s^k \rangle} \sigma_{\langle X_{s,h}^k \rangle}} \\
&= \frac{E[(X_s^k - \mu_{X_s^k})(X_{s,h}^k - \mu_{X_{s,h}^k})]}{\sigma_{X_s^k} \sigma_{X_{s,h}^k}} \\
&= C(X_s^k, X_{s,h}^k).
\end{aligned} \tag{21}$$

Likewise, we obtain

$$C(\langle X_s^k \rangle, \langle X_{s,v}^k \rangle) = C(X_s^k, X_{s,v}^k). \tag{22}$$

In view of (19), (21) and (22), it follows

$$R(X_s^k, X_{s,h}^k, X_{s,v}^k) = R(\langle X_s^k \rangle, \langle X_{s,h}^k \rangle, \langle X_{s,v}^k \rangle). \tag{23}$$

□

The measure of cross-correlation R is used to look for a candidate pair of keys among 128 pairs because for each possible key, α_s , its flipped version, $\alpha_{\langle s \rangle}$, corresponds to the same value of R (see *Proposition 1*). Interestingly, if α_s is the correct key and a different key $\alpha_{s'}$ was used, except its flipped version, the decrypted images would show a clear decrease in cross-correlation. Indeed, we have

$$X_{s'}^k(i, j) = X_s^k(i, j) + d_{s'}^k(i, j), \tag{24}$$

where $d_{s'}^k$ represents the distortion caused by the decryption with $\alpha_{s'}$. Under the assumption that $d_{s'}^k$ is independent of X_s^k , it can be shown that

$$R(X_{s'}^k, X_{s',h}^k, X_{s',v}^k) < R(X_s^k, X_{s,h}^k, X_{s,v}^k). \tag{25}$$

We provide the proof below.

Proof. The correlation coefficient between the distorted image $X_{s'}^k$ and its shifted version in the horizontal direction can be given as

$$\begin{aligned}
C(X_{s'}^k, X_{s',h}^k) &= \frac{E[(X_{s'}^k - \mu_{X_{s'}^k})(X_{s',h}^k - \mu_{X_{s',h}^k})]}{\sigma_{X_{s'}^k} \sigma_{X_{s',h}^k}} \\
&= \frac{E[(X_s^k - \mu_{X_s^k} + (d_{s'}^k - \mu_{d_{s'}^k}))(X_{s,h}^k - \mu_{X_{s,h}^k} + (d_{s',h}^k - \mu_{d_{s',h}^k}))]}{\sigma_{X_s^k + d_{s'}^k} \sigma_{X_{s,h}^k + d_{s',h}^k}}.
\end{aligned} \tag{26}$$

Under the assumption that $d_{s'}^k$ is independent of X_s^k , we obtain

$$C(X_{s'}^k, X_{s',h}^k) \approx \frac{E[(X_s^k - \mu_{X_s^k})(X_{s,h}^k - \mu_{X_{s,h}^k})]}{\sqrt{(\sigma_{X_s^k}^2 + \sigma_{d_{s'}^k}^2)(\sigma_{X_{s,h}^k}^2 + \sigma_{d_{s',h}^k}^2)}}. \quad (27)$$

Since $\sqrt{(\sigma_{X_s^k}^2 + \sigma_{d_{s'}^k}^2)(\sigma_{X_{s,h}^k}^2 + \sigma_{d_{s',h}^k}^2)} > \sigma_{X_s^k} \sigma_{X_{s,h}^k}$, we get $C(X_{s'}^k, X_{s',h}^k) < C(X_s^k, X_{s,h}^k)$. Similarly, one can deduce that $C(X_{s'}^k, X_{s',v}^k) < C(X_s^k, X_{s,v}^k)$ which eventually leads to $R(X_{s'}^k, X_{s',h}^k, X_{s',v}^k) < R(X_s^k, X_{s,h}^k, X_{s,v}^k)$. \square

This means there is only one pair of possible keys which correspond to the highest cross-correlation, measured here by R . The candidate pair of keys consists of the correct key α_s and its flipped version $\alpha_{\langle s \rangle}$. To illustrate this point practically, a key α_s has been created so that the sequence s is equal to 219 in decimal. Then, a number of images have been encrypted using α_s but decrypted using all the possible keys $\alpha_{s'}$ with $s' = \{0, 1, \dots, 255\}$. Fig. 7 shows the values of R for different decrypted images. As can be seen, the pair $(36, 219)$ always corresponds to the highest cross-correlation

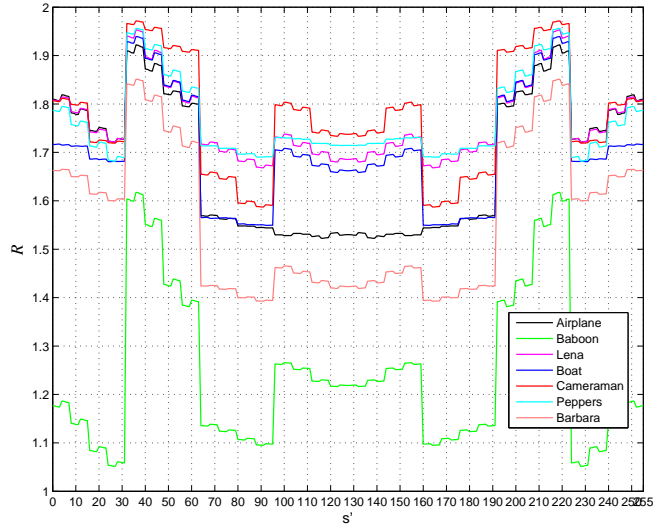


Figure 7: Cross-correlation of decrypted images with 256 possible keys. The cross-correlation measure takes its highest value at the pair $(36, 219)$ which corresponds to the correct key and its flipped version.

when compared to other possible keys. Note that the key α_{36} is the flipped version of the correct key, i.e., $36 = 255 - 219$ and this is in perfect agreement with *Proposition 1*. Given 256 possible keys, the candidate pair of keys $(\alpha_{s^*}, \alpha_{\langle s^* \rangle})$ is selected so that

$$s^* = \arg \max_{\epsilon \in \{0, 1, \dots, 255\}} \{R(X_\epsilon^k, X_{\epsilon,h}^k, X_{\epsilon,v}^k)\}. \quad (28)$$

Once the candidate pair of keys is determined, the attacker can easily distinguish the correct one from its flipped version because the visual effect of the flipped version on the decrypted image is clearly noticeable. This is illustrated by Fig. 8 where a number of images decrypted with the correct key and its flipped version are displayed. Finally, the proposed algorithm for estimating the key can be summarized as follows.

1. **Initialization:** For $q = \{0, 1, \dots, 7\}$, initialize the top left corner bit $\hat{\Phi}_q(1, 1) = 0$.



Figure 8: Visual difference between the images decrypted with the correct key and those decrypted with its flipped version. First row: decryption using the correct key. Second row: decryption using the flipped version of the correct key.

2. Estimation: Set $q = 7$ and do

- (a) *Horizontal estimation:* Estimate the first row of the key using (9).
- (b) *Vertical estimation:* Estimate the remaining bits in the vertical direction as described by (11).
- (c) Set $q = q - 1$. If $q \geq 0$ go to step a).

3. Refinement:

- (a) Derive 255 possible keys from of the current key by flipping its bit planes $\hat{\Phi}_q, q = \{0, 1, \dots, 7\}$, accordingly.
- (b) Use each of the 256 keys to decrypt an image and select the candidate pair of keys that gives the highest cross-correlation measure using (28)
- (c) Display the two decrypted images obtained with the candidate pair of keys and select the correct one according to the visual difference.

4. Experimental analysis

In the experimental analysis, standard gray level images of size 512×512 are used. These images were obtained from [39, 40]². It was claimed in section 2 that the attacker only needs a partial access to the key in order to decipher encrypted images. In the first set of experiments, we assess the influence of the least significant bit planes (i.e., $q \in \{0, 1, 2\}$) on the reconstruction quality. For the sake of illustration, Fig. 9 shows the reconstruction of four standard images where the least three significant bit planes of the key Φ_q are randomly generated. As can be seen, the images can be reconstructed with good visual quality and unnoticeable distortions.

To analyze the performance of the proposed algorithm, a number of images have been encrypted using the stream cipher with a randomly generated key Φ_q where $q = \{0, 1, \dots, 7\}$. The proposed COA has been applied to estimate the key and disclose the content of encrypted images. It is worth mentioning that the estimation of the key depends on

²The images available in [39] are cropped to 512×512 .

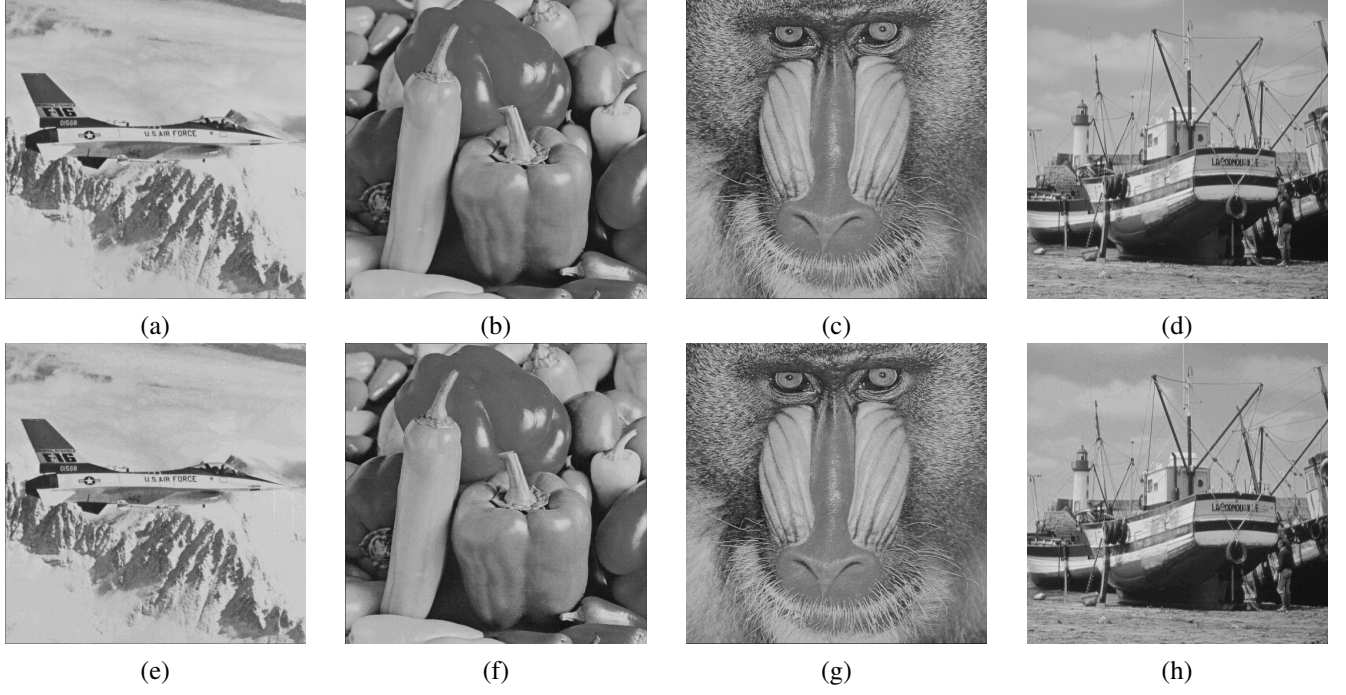


Figure 9: Reconstruction of standard images using a random key in the least three significant bit planes. (a) Original Airplane. (b) Original Peppers. (c) Original Baboon. (d) Original Boat. (e) Reconstructed Airplane with PSNR=35.73 dB. (f) Reconstructed peppers with PSNR=35.69 dB. (g) Reconstructed Baboon with PSNR=35.70 dB. (h) Reconstructed Boat with PSNR=35.69 dB.

the number of encrypted images, ℓ , that are accessible to the attacker. This is referred to as the unicity distance in the literature [33, 34]. Denote by Φ_q the actual key and $\hat{\Phi}_q$ its estimate. The correct estimation bit rate for each bit plane significance level q of the encryption key is measured in percentage as

$$\beta(q) = \frac{100}{M N} \times \sum_{i=1}^M \sum_{j=1}^N \left(1 - |\Phi_q(i, j) - \hat{\Phi}_q(i, j)| \right). \quad (29)$$

Fig. 10 plots the correct bit rate in each plane against the number of observed images. As expected, the estimation of the most significant bit planes requires less images than that of the least significant bit planes. Interestingly, 480 encrypted

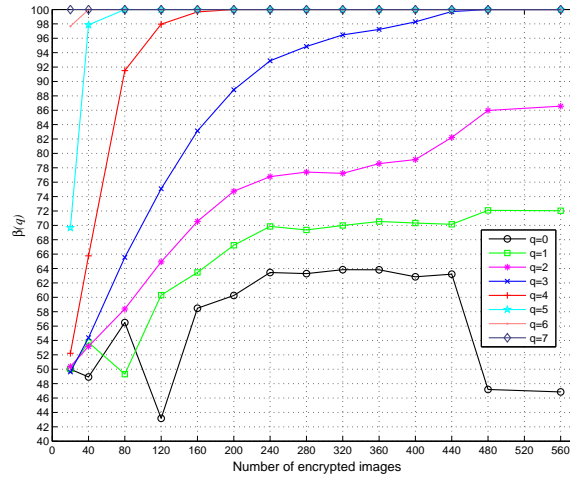


Figure 10: Bite rate of the correct estimation of the encryption key for each bit plane significance level.

images are sufficient to estimate the 5 most significant bit planes without errors, i.e., $\beta(q) = 100\%$ for $q \geq 3$. Note also that the third least significant bit plane (i.e., $q = 2$) is estimated with an accuracy higher than 86% whereas the second least significant bit plane is estimated with more than 72% of accuracy. This estimation enables the attacker to reconstruct images with high visual quality. From the results, it can be seen that the three most significant bit planes, $q \in \{5, 6, 7\}$, can be accurately estimated with 80 images only. As expected, the least significant bit plane of the key (i.e., $q = 0$) cannot be estimated because natural images do not exhibit any inter-pixel redundancy (see Fig. 4 and Fig. 5 for $q = 0$). Fig. 11 illustrates the visual quality of the reconstructed 'peppers' for different values of ℓ and lists the corresponding Peak Signal to Noise Ratio (PSNR), accordingly. It is worth mentioning that similar results were also obtained on other images with

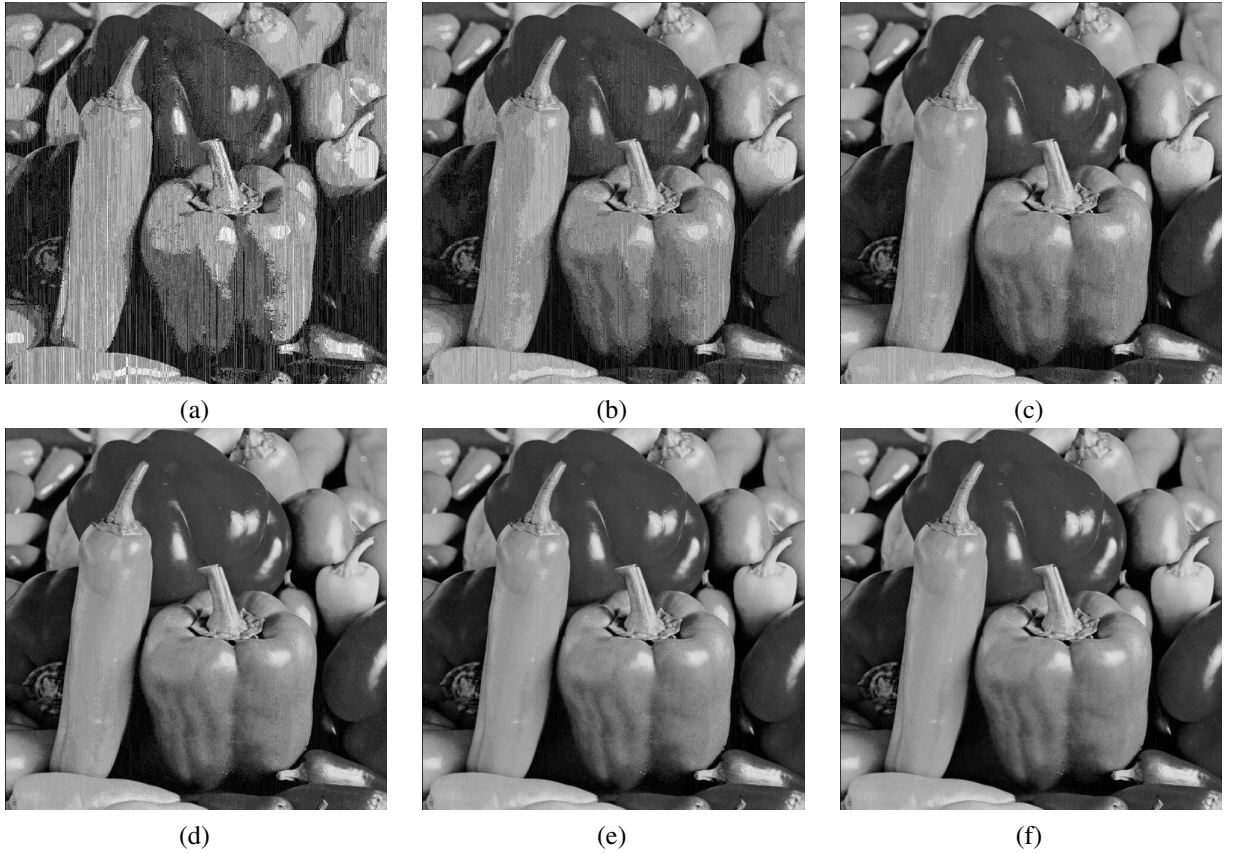


Figure 11: Decryption of the encrypted 'Peppers' with a key estimated for different values of ℓ . (a) $\ell = 20$, PSNR=20.47 dB. (b) $\ell = 40$, PSNR=26.32 dB. (c) $\ell = 80$, PSNR=30.91 dB. (d) $\ell = 160$, PSNR=35.59 dB. (e) $\ell = 320$, PSNR=39.42 dB. (f) $\ell = 480$, PSNR=42.23 dB.

minor differences in PSNR. Overall, with 480 encrypted images, the reconstructed visual quality was above 42 dB in terms of the PSNR.

5. Proposed solutions

It is clear from the findings presented earlier that the security of the stream cipher can be compromised on natural images because of the inter-pixel redundancy that characterizes such data. In fact, the spatial redundancy which is normally exploited by the content owner to recover the image losslessly can also be used by the data hider to estimate the encryption key. One straightforward solution is to use a different cipher, e.g. a block cipher such as the Advanced Encryption

Standard (AES) that has been proven to be secure with a key length of 128 or 256 bits. However, since the stream cipher has been adopted in the literature for its simplicity and suitability for high capacity data hiding, other solutions could be suggested while keeping the same cipher.

5.1. Pseudo-random pixel permutation

A possible solution to enhance the security of existing reversible data hiding techniques in the encrypted domain consists of pseudo-randomly permuting pixels before encryption. This, however, may not work out with some systems since their design requires the same pixel arrangement for data hiding in order to recover the lossless version of the image. Indeed, the data hiding process in [11], [13], and [18] divides the pixels of each block in two sets and flips the three least significant bit planes of a set according to the value of the additional bit. Because the data owner uses spatial correlation in each block to recover the image while the data hider uses it for extracting the additional data, the pixels cannot be permuted prior to data hiding. Similarly, the data hiding process in [17] requires the original image structure to select the pixels that carry the additional data among other neighboring pixels via the hiding key. On the contrary, the techniques reported in [14], [15], and [16] can be enhanced by pseudo-randomly shuffling pixels at the encryption stage because neither the image recovery stage nor the data hiding process relies on inter-pixel redundancy. The data provider in [12] reserves room prior to stream encryption by self-embedding the least significant bit planes of a part A of the image into another part B. This is to allow the data hider to use the room created in part A for embedding the additional data. If the pixels in part A and B are pseudo-randomly shuffled with a secret key before applying the stream cipher, the system becomes more secure and robust against the COA. In [21], both the data owner and the data hider use a stream cipher where the data hiding key is made public. While the data owner uses statistical features to distinguish original blocks from encrypted ones through a double decryption process, image pixels can be permuted with another secret key before encryption to combat the COA without affecting the data hiding process. Finally, the patches used in [26] could be shuffled with a secret key at the encryption stage in order to avoid the estimation of the key via COA especially in those edged and textured areas where the sparse coding is not applied.

5.2. Image-dependent key stream

It is clear that the reuse of the key stream to encrypt multiple images of different contents contributes to the estimation of the key. To overcome this issue, one can adjust the application of RDH in encrypted images by changing the key stream every time an image is presented. Indeed, an image-dependent key stream will prevent the attacker from estimating the key. Denote by Φ_q^k the key stream corresponding to image X^k at the significance level q . Then, (3) becomes

$$\bar{b}_q^k(i, j) = b_q^k(i, j) \oplus \Phi_q^k(i, j). \quad (30)$$

To implement the above amendment, the content owner may keep a number of bits, selected from the image, in *unencrypted* form. Since these bits should be uniquely characterizing the image content, one can make use of bits in the least

significant bit plane as it offers a highly random distribution of bits with equal probability. The selected bits will then be combined with a secret key to derive a seed for generating the key stream. Once generated, the key stream can be used to encrypt the remaining bits using (30). It is worth noting that the number of bits selected to generate the image-dependent key stream (Part A) is negligible when compared to the number of encrypted bits (Part B). This maintains the data hiding capacity of existing state-of-the-art systems while enhancing their security.

6. Conclusion

Reversible data hiding in encrypted images has increasingly attracted many researchers in the field of information processing in the encrypted domain for cloud computing and privacy-preserving applications. This is an effective approach to set up a secure communication between the content owner (data provider) and the data hider (service provider). The data hider receives the host image in encrypted form and embeds/extracts additional data without accessing the visual content of the host image whereas the content owner recovers the host data losslessly while maintaining its confidentiality through encryption. The reversibility requirement on data hiding in the encrypted domain is normally met by exploring the change in spatial correlation caused by the data hiding process. This justifies the use of a stream cipher in most state-of-the-art systems since the spatial arrangement of pixels in an image remains unchanged. However, this could also allow the data hider to exploit inter-pixel redundancy and compromise the security of the encryption system. Unlike existing work, this paper is the first one that analyzes RDH in encrypted images from a security perspective where the data hider acts as the attacker. We proposed a Ciphertext-Only Attack and highlighted the weakness of current state-of-the-art data hiding systems that adopted a stream cipher to protect privacy from the data hider. In the proposed attack, the data hider who acts as the adversary exploits inter-pixel redundancy to estimate the encryption key with high accuracy when it is reused several times on images of different contents. Experiments have validated the proposed technique and showed that the visual content of encrypted images can be disclosed efficiently. Finally, possible solutions have been provided to tackle the security issue with RDH schemes in encrypted images. This allows researchers to consider the security aspect of their systems in the future and develop more robust techniques.

References

- [1] M. Goljan, J. Fridrich, and R. Du. Invertible authentication. In *Proc. International Workshop on Information Hiding*, pages 27–41, Pittsburgh, PA, Apr. 2001.
- [2] Z. Ni, Y-Q. Shi, N. Ansari, and W. Su. Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16:354–362, Mar. 2006.
- [3] X. Li, W. Zhang, X. Gui, and B. Yang. Efficient reversible data hiding based on multiple histograms modification. *IEEE Transactions on Information Forensics and Security*, 10:2016–2027, Sep. 2015.

- [4] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran. On compressing encrypted data. *IEEE Transactions on Signal Processing*, 52:2992–3006, Oct. 2004.
- [5] W. Liu, W. Zeng, L. Dong, and Q. Yao. Efficient compression of encrypted grayscale images. *IEEE Transactions on Image Processing*, 19:1097–1102, Apr. 2010.
- [6] S. Lian, Z. Liu, Z. Ren, and H. Wang. Commutative encryption and watermarking in video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 17:774–778, Jun. 2007.
- [7] T. Bianchi, A. Piva, and M. Barni. On the implementation of the discrete Fourier transform in the encrypted domain. *IEEE Transactions on Information Forensics and Security*, 4:8697, Mar. 2009.
- [8] T. Bianchi, A. Piva, and M. Barni. Composite signal representation for fast and storage-efficient processing of encrypted signals. *IEEE Transactions on Information Forensics and Security*, 5:180187, Mar. 2010.
- [9] Z. Qian, X. Zhang, and S. Wang. Reversible data hiding in encrypted JPEG bitstream. *IEEE Transactions on Multimedia*, 16:1486–1491, Aug. 2014.
- [10] W. Puech, M. Chaumont, and O. Strauss. A reversible data hiding method for encrypted images. In *Proc. SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents*, volume X-68191E, San Jose, CA, USA, Feb. 2008.
- [11] X. Zhang. Reversible data hiding in encrypted image. *IEEE Signal Processing Letters*, 18:255–258, Apr. 2011.
- [12] K. Ma, W. Zhang, X. Shao, N. Yu, and F. Li. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, 8:553–562, Mar. 2013.
- [13] W. Hong, T.-S. Chen, and H.-Y. Wu. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Processing Letters*, 19:199–202, Apr. 2012.
- [14] X. Zhang. Separable reversible data hiding in encrypted image. *IEEE Transactions on Information Forensics and Security*, 7:826–832, Apr. 2012.
- [15] X. Zhang, Z. Qian, G. Feng, and Y. Ren. Efficient reversible data hiding in encrypted images. *Journal of Visual Communications and Image Representation*, 25:322–328, 2014.
- [16] Z. Qian and X. Zhang. Reversible data hiding in encrypted image with distributed source encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 26:636–646, Apr. 2016.
- [17] X. Wu and W. Sun. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Processing*, 104:387–400, 2014.

- [18] C. Qin and X. Zhang. Effective reversible data hiding in encrypted image with privacy protection for image content. *Journal of Visual Communications and Image Representation*, 31:154–164, 2015.
- [19] Z. Qian, X. Zhang, Y. Ren, and G. Feng. Block cipher based separable reversible data hiding in encrypted images. *Multimedia Tools and Applications*, 75:13749–13763, July 2016.
- [20] Y.-C. Chen, C.-W. Shiu, and G. Horng. Encrypted signal-based reversible data hiding with public key cryptosystem. *Journal of Visual Communications and Image Representation*, 25:1164–1170, 2014.
- [21] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, and Y. Y. Tang. Secure reversible image data hiding over encrypted domain via key modulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 26:441–452, Mar. 2016.
- [22] Z. Qian, X. Zhang, and G. Feng. Reversible data hiding in encrypted images based on progressive recovery. *IEEE Signal Processing Letters*, 23:1672–1676, Nov. 2016.
- [23] Z. Qian, S. Dai, F. Jiang, and X. Zhang. Improved joint reversible data hiding in encrypted images. *Journal of Visual Communications and Image Representation*, 40:732–738, 2016.
- [24] W. Zhang, K. Ma, and N. Yu. Reversibility improved data hiding in encrypted images. *Signal Processing*, 94:118–127, 2014.
- [25] D. Xu and R. Wang. Separable and error-free reversible data hiding in encrypted images. *Signal Processing*, 123:9–21, 2016.
- [26] X. Cao, L. Du, X. Wei, D. Meng, and X. Guo. High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Transactions on Cybernetics*, 46:1132–1143, May. 2016.
- [27] S. Li, C. Li, K.-T. Lo, and G. Chen. Cryptanalysis of an image scrambling scheme without bandwidth expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:338–349, Mar. 2008.
- [28] M. Bertilsson, E. F. Brickell, and I. Ingemarson. Cryptanalysis of video encryption based on space-filling curves. In *Proc. Advances in Cryptology (Lecture Notes in Computer Science)*, volume 434, pages 403–411, Berlin, Germany, Aug. 1989.
- [29] W. Li, Y. Yan, and N. Yu. Breaking row-column shuffle based image cipher. In *Proc. the ACM International Conference on Multimedia (MM)*, pages 1097–1100, New York, NY, USA, Oct. 2012.
- [30] S. Li, C. Li, G. Chen, N. G. Bourbakis, and K.-T. Lo. A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing: Image Communication*, 23(3):212–223, 2008.
- [31] C. Li and K.-T. Lo. Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing*, 91(4):949–954, 2011.

- [32] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy. On the security of permutation-only image encryption schemes. *IEEE Transactions on Information Forensics and Security*, 11:235–246, Feb. 2016.
- [33] Y. Mao and M. Wu. Unicity distance of robust image hashing. *IEEE Transactions On Information Forensics and Security*, 2:462–467, Sep. 2007.
- [34] F. Khelifi and J. Jiang. Analysis of the security of perceptual image hashing based on non-negative matrix factorization. *IEEE Signal Processing Letters*, 17:43–46, Jan. 2010.
- [35] L. Perez-Freire, F. Perez-Gonzalez, T. Furon, and P. Comesana. Security of lattice-based data hiding against the known message attack. *IEEE Transactions on Information Forensics and Security*, 1:421–439, Dec. 2006.
- [36] A. Biryukov and E. Kushilevitz. *Advances in cryptology*, chapter From differential cryptanalysis to ciphertext-only attacks, pages 72–88. Lecture notes in Computer Science. Springer, 1998.
- [37] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 5th edition, 2001.
- [38] D. P. Francis, A. J. S. Coats, and D. G. Gibson. How high can a correlation coefficient be? effects of limited reproducibility of common cardiological measures. *International Journal of Cardiology*, 69:185–189, 1999.
- [39] <http://imagedatabase.cs.washington.edu/groundtruth/>. *Image retrieval dataset, Efficient Content-based Retrieval Group, University of Washington*. accessed in 2009.
- [40] <http://decsai.ugr.es/cvg/dbimagenes>. *Standard test images, Computer Vision Group, University of Granada*. accessed in 2009.